

MMDOS

Síťové analýzy PID

Michael Kala a Petra Mariana Millarová

únor 2019

Konzolová aplikace, která provádí síťové analýzy nad daty PID za využití pgRouting extenze PostGISu

- vstup - adresy počátečního a koncového bodu
- výstup - nejkratší cesta MHD spolu s doporučenými přestupy

Zdroje dat

- Adresní místa RÚIAN Hlavního města Prahy - nahlížení do KN
 - ▶ formát *.csv
- Trasy linek PID - portál opendata Hlavního města Prahy
 - ▶ formát *.shp
- Zastávky PID - portál opendata Hlavního města Prahy
 - ▶ formát *.shp

Zpracování dat

- tvorba tabulky a geometrie pro adresní místa
- převod dat PID ze *.shp do tabulek
- odstranění nočních linek z dat PID
- přetypování sloupce zast_uzel

Topologie - hrany

- přiřazení správných zast_uzel do sloupců source a target pomocí skriptu

```
SELECT DISTINCT ON(t.gid) t.gid, z.zast_uzel_  
FROM trasy t, zastavky z  
WHERE ST_DWithin(ST_StartPoint(ST_LineMerge(t.geom)), z.geom, 500)  
AND t.zast_id_od = z.zast_id
```

- vytvoření tabulky uzlů pomocí funkce `pgr_createVerticesTable` a přejmenování sloupce `the_geom` na `geom`

```
SELECT pgr_createVerticesTable('trasy', 'geom', 'source', 'target');  
ALTER TABLE trasy_vertices_pgr RENAME COLUMN the_geom TO geom;
```

Konzolová aplikace - pomocné SQL funkce

```
FindStationName(id INTEGER)  
FindVertexID(cd INTEGER, co INTEGER, u VARCHAR)  
FindVertexIDcd(cd INTEGER, u VARCHAR)  
FindVertexIDori(co INTEGER, u VARCHAR)  
FindVertexIDst(u VARCHAR)
```

Konzolová aplikace - ukázka funkce

- Nalezení nejbližšího uzlu k zadané adrese

```
CREATE OR REPLACE FUNCTION
```

```
    FindVertexID(cd INTEGER, co INTEGER, u VARCHAR)
```

```
RETURNS INTEGER AS $id$
```

```
declare
```

```
    id INTEGER;
```

```
BEGIN
```

```
    SELECT v.id INTO id FROM adr a, trasy_vertices_pgr v
```

```
    WHERE a.c_domovni = cd
```

```
    AND a.c_orientacni = co
```

```
    AND a.ulice = u
```

```
    ORDER BY (a.geom)<->(v.geom) ASC LIMIT 1;
```

```
    RETURN id;
```

```
END;
```

```
$id$ LANGUAGE plpgsql;
```


Konzolová aplikace - výpočet nejkratší trasy

- Dijkstrův algoritmus

```
SELECT z.zast_nazev, trasy.l_metro, trasy.l_tram, trasy.l_bus, trasy.l_lan ,  
        trasy.l_vlak, trasy.l_lod, node  
FROM pgr_dijkstra(  
    'SELECT gid as id, source, target, CAST(shape_leng as REAL) AS cost  
    FROM trasy', id_from, id_to)  
JOIN (  
    SELECT DISTINCT(zastavky.zast_uzel_), zastavky.zast_nazev  
    FROM zastavky  
    ) AS z ON node = z.zast_uzel_  
LEFT JOIN trasy ON edge = trasy.gid  
ORDER BY seq;
```

Konzolová aplikace - výpis zastávek

- trasy - v každém úseku údaje o všech linkách, které tudy vedou
- průchod nejkratší trasou a výběr linek s nejmenším počtem přestupů:
 - 1 průchod všech linek projíždějících první zastávkou, výpočet dosahu každé linky
 - 2 přiřazení čísla linky s nejdelším dosahem
 - 3 zopakování kroků 1 a 2

Konzolová aplikace - ukázka výstupu

```
Connecting to database pgis_uzpd on geo102.fsv.cvut.cz...
Connected!
```

```
-----
MMDOS ©
```

```
Created by: Maru & Michael 2019
-----
```

```
Start with typing from where you want to travel:
```

```
Street: Evropská
```

```
House number: 612
```

```
House orientation number:
```

```
Where do you want to go?
```

```
Street: Thákurova
```

```
House number:
```

```
House orientation number: 7
```

```
Your route from Evropská 612/79 to Thákurova 2077/7:
```

Stop	Line
Sídlíště Červený Vrch	Get on: 20
Bořislavka	20
Na Pískách	20
Hadovka	20
Thákurova	Transfer to: 108
Studentský dům	Get off here!

```
Again? (Y/N): n
```

Závěr - problémy a jejich řešení

- stejné "unikátní" ID v tabulce zastavky
- kombinace nočních a denních spojů
- ocenění tras - v reálu je trasa s méně přestupy rychlejší, než nejkratší trasa s více přestupy

Děkujeme za pozornost.